

## **Análisis e interpretación de los verbos auxiliares del español en un sistema NLQ**

Antonio Moreno Sandoval  
Cristina Olmeda Moreno

Dpto. de Lingüística y Lógica  
Universidad Autónoma de Madrid  
Centro de Investigación UAM - IBM

---

### **Resumen**

Esta comunicación presenta un tratamiento completo de los verbos auxiliares *ser*, *estar* y *haber*, así como de las construcciones oracionales en las que pueden aparecer. El tratamiento incluye tanto el análisis como la interpretación de este tipo de oraciones, pero restringidas al ámbito de la interrogación de bases de datos en lengua natural (NLQ). Esto excluye el uso de fórmulas idiomáticas, la distinción entre *ser* y *estar* o la interpretación arcaica de *haber* como verbo de posesión.

Se basa, por tanto, en criterios semánticos y pragmáticos, dando al comportamiento sintáctico una importancia secundaria. Nuestro tratamiento se caracteriza por la interpretación unitaria de *ser* y *estar* y la especialización de *hay* dentro de su paradigma etimológico. Los tres verbos (*ser*, *estar*, *hay*) se interpretan como una proposición con cuantificador existencial. Este sistema, que está desarrollando el Dpto. de Lingüística de la Universidad Autónoma de Madrid para el Centro de Investigación UAM-IBM, utiliza PROLOG.

# Análisis e interpretación de los verbos auxiliares del español en un sistema NLQ

## Breve introducción al sistema

La finalidad que persigue el proyecto que estamos desarrollando es interrogar a una base de datos utilizando el castellano. Esto se conoce en la literatura científica como *Natural Language Query* (NLQ) o también *natural language front ends* (Gazdar y Mellish, 1989). Se trata en realidad de una interfaz entre el usuario y la base de datos que interpreta la pregunta realizada en una lengua natural y traduce a un lenguaje específico de bases de datos (en nuestro caso, SQL). Esta estrategia de consulta presenta las ventajas de no necesitar una formación especial en bases de datos (cualquier usuario puede realizar la consulta directamente) y, por otro lado, es más flexible que los sistemas de menús, pues permite hacer un número bastante más amplio de preguntas distintas.

Este sistema es una parte integrante de otro más general, que se encargará de buscar en la base de datos y responder al usuario, ya sea en forma de simples afirmaciones ('Sí', 'No', '5', 'en Enero de 1990'... si se trata de interrogaciones totales o con elementos QU, 'cuántos...', 'cuándo') o bien proporcionando listas y gráficos (si el usuario hizo una petición concreta: 'quiero nombres de ...', 'los alumnos de 5º curso'). A estos sistemas generales se les suele denominar *Question Answering Systems* (QA). No hace falta destacar el atractivo que puede tener para un usuario de una base de datos el contar con un sistema de este tipo. Veamos sucintamente las características esenciales que debe tener un sistema QA:

1. Analizar e interpretar cualquier pregunta o petición, dentro de un fragmento limitado de un lenguaje natural y del contexto general de lo que se puede preguntar a una base de datos. A esta cobertura lingüística le dedicaremos un pequeño apartado.
2. Traducir la interpretación, que estará formulada en un lenguaje de representación de conocimiento, a un lenguaje de bases de datos. El modelo conceptual codificado por dicho lenguaje de representación del conocimiento tiene que ajustarse a todos los aspectos relevantes de la aplicación concreta (una base de datos de una empresa, de una biblioteca, de contenido informativo, etc.).
3. Permitir al usuario que compruebe que el sistema ha entendido su pregunta. Esto se puede conseguir o bien mediante la presentación de la interpretación en el lenguaje de representación conceptual (que suele tener la apariencia de una fórmula lógica), o bien a través de una paráfrasis en la misma lengua (esto requiere un generador que tome la representación y la convierta en una secuencia gramatical, aunque no necesariamente tiene que ser igual que la oración original). En el último caso, si el sistema no proporciona la comprobación sólo queda esperar la respuesta. Esta característica es, por tanto, opcional pero muy útil, pues evita búsquedas y esperas infructuosas. No olvidemos que incluso oraciones triviales presentan algún tipo de ambigüedad, que se refleja luego en la interpretación. Ninguna interpretación conceptual debería ser ambigua. Escoger la interpretación adecuada o la que más nos interesa ahorra tiempo.
4. La respuesta a la consulta del usuario tiene que ajustarse a lo que él o ella esperan, es decir, una lista, un gráfico, una respuesta afirmativa o negativa, etc.

La parte que concierne a nuestra investigación son los puntos 1 y 3, es decir el analizador de cadenas<sup>1</sup> (supuestamente gramaticales) en castellano y el generador de paráfrasis en castellano a partir de una forma lógico-conceptual.

El analizador proporciona una o varias estructuras sintácticas por cadena, cada una con su estructura semántica. Está basado en el *User Specialty Languages System* (USL) desarrollado por IBM (Zoepprit

384), aunque con un formalismo nuevo compilable en Prolog. El análisis morfológico está subsumido dentro del sintáctico mediante unas extensiones especiales en las reglas sintácticas para reconocer determinados tipos de cadenas. Análogamente, las reglas sintácticas tienen otras 'extensiones' acerca de la naturaleza semántica del constructo que se está construyendo. De esta forma, cuando se aplican las reglas de la gramática se está procesando al mismo tiempo información morfológica, sintáctica y semántica. Esto permite mucha flexibilidad a la hora de hacer reglas *ad hoc*, y aunque no es muy legante sí es eficaz.

Por otra parte, existen dos modelos conceptuales: uno con el vocabulario básico necesario para construir oraciones en castellano, es decir, el conjunto de categorías cerradas de la lengua; el otro con vocabulario específico de la aplicación. El primero de estos modelos está restringido por el dominio de la lengua que comprende un sistema NLQ. El segundo está condicionado por la estructura de la base de datos a la que se aplica.

El sistema presenta la siguiente organización:

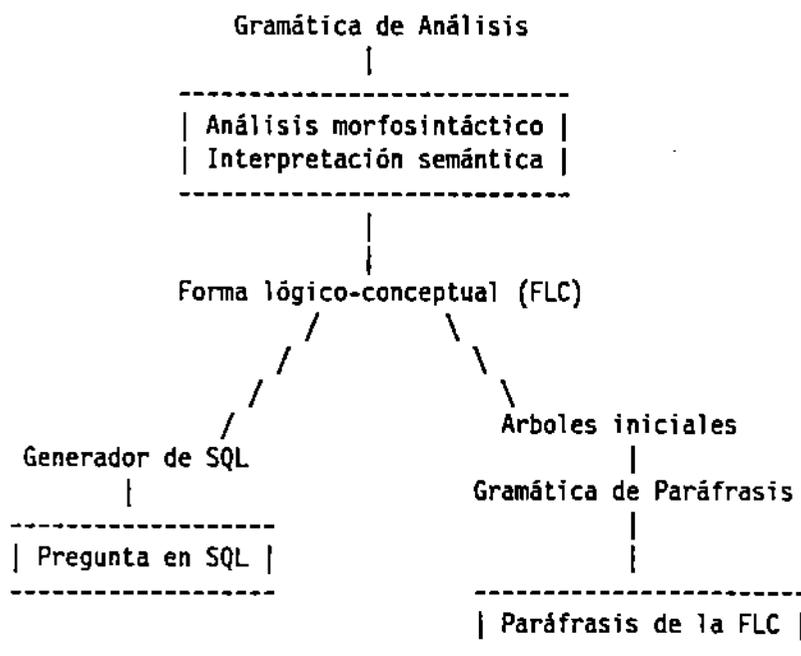


Figura 1. Niveles del sistema

## Cobertura lingüística

El sistema se limita a interpretar el subconjunto del castellano que se ha considerado relevante para la interrogación de bases de datos. Este subconjunto comprende las siguientes construcciones:

- *Oraciones imperativas*: contienen una "orden" a la base de datos. Por ejemplo:

Dame una lista de empleados

- *Oraciones interrogativas directas*:

- *Interrogaciones totales*: su función es verificar si la información contenida en la pregunta es cierta o no. Se responden con un "sí" o un "no".

¿ Trabaja Gómez en el departamento de ventas ?

- *Interrogativas parciales o de constituyentes*: son preguntas que contienen pronombres interrogativos. La respuesta es una lista o tabla que proporciona la información representada por el constituyente interrogado en la oración.

- *Oraciones interrogativas indirectas*: en este tipo se incluyen tanto las totales como las parciales. Su interpretación es la misma que la de las interrogativas directas.

Quiero saber cuántos empleados hay en el departamento.

Quiero saber si hay empleados casados en el departamento.

- *Oraciones relativas*: pueden aparecer como constituyente dentro de cualquiera de los tipos anteriores.

Dame los empleados que trabajan con Pérez

¿ Cuáles son los productos que exportamos a Francia ?

Otros procesos sintácticos que se tratan son la coordinación, las construcciones comparativas y ciertos tipos de pronominalización.

---

## Verbos auxiliares y verbos plenos

La distinción entre verbos auxiliares y verbos plenos en un sistema como el que nos ocupa no está basada en las tradicionales distinciones entre verbos con 'significado' y verbos 'vacíos'<sup>2</sup>. Tampoco la naturaleza del predicado (verbos predicativos y verbos atributivos) es un factor determinante. El criterio que empleamos se basa en la naturaleza de las entidades en las bases de datos. Consideramos *verbos plenos* aquellos que establecen relaciones entre entidades. Imaginémonos, por ejemplo, que tenemos una base de datos de geografía económica, donde la relación **exporta** está representada en la base de datos por tres listas de **exportadores**, **productos** y **compradores**:

exportadores	productos	compradores
Arabia Saudí	petróleo	CEE
Colombia	café	CEE
España	naranjas	Alemania Federal
...	...	...

El verbo 'exportar' no existe como tal en la base de datos, pero sí en el modelo conceptual como un predicado Prolog que relaciona estos tres argumentos

```
exporta(Exportadores,Productos,Compradores).
exporta(colombia,"café",cee).
```

Por lo tanto, el contenido del verbo no interesa en sí mismo, 'exportar' es simplemente un funtor que une distintos argumentos.

En cambio, consideramos *verbos auxiliares* a aquellos a los que podemos asignar una interpretación determinada e independiente de las entidades que intervengan en la relación. En consecuencia, y paradójicamente, en nuestro sistema los verbos auxiliares tienen un contenido propio (más adelante veremos cuál es), mientras que los verbos plenos son meramente relacionales. Desde este punto de vista, lo más adecuado sería distinguir entre *verbos con interpretación definida por el sistema* (los verbos auxiliares) y *verbos con interpretación definida por el usuario* (los verbos plenos). Por motivos prácticos seguiremos usando la terminología tradicional.

## Características de los verbos auxiliares

Antes de proseguir en la exposición, es necesario destacar que estos verbos sintácticamente pueden realizar una función auxiliar o principal. Sólo cuando ocupan una posición principal en la oración reciben la interpretación definida por el sistema. Si aparecen como elementos en las formas compuestas o en la pasiva son tratados por las reglas morfosintácticas correspondientes pero, evidentemente, ninguna información semántica se filtra al constructo creado. La siguiente tabla muestra las diferentes posibilidades de los tres verbos:

	Función auxiliar	Función principal
<b>Ser</b>	<i>pasiva</i>	<i>cópula; con locativos; con temporales; con sintagmas preposicionales</i>
<b>Estar</b>	<i>pasiva, perífrasis de gerundio</i>	<i>cópula; con locativos; con temporales; con sintagmas preposicionales</i>
<b>Haber</b>	<i>tiempos compuestos</i>	<i>existencial (hay)</i>

Tabla 1. Verbos auxiliares

La tabla nos permite apreciar las principales coincidencias y diferencias entre estos verbos. Por una parte, la pareja 'ser'-'estar' presenta una misma distribución funcional, aunque con distinta interpretación, cuando son verbos principales de la oración. Veremos que las diferencias semánticas entre 'ser' y 'estar' no son relevantes en el contexto NLQ. En cambio, funcionando sintácticamente como auxiliares manifiestan una distinción crucial: 'estar' es el único que puede aparecer en las perífrasis de gerundio (*está leyendo; \*es leyendo*). En resumen, 'ser' y 'estar' como verbos principales reciben la misma interpretación; como verbos auxiliares se tratan de distinta forma para distinguirlos en los contextos sintácticos pertinentes.

Por otra parte, 'haber' presenta una distinción interna: las formas 'auxiliares' y las formas 'existenciales'. Las primeras corresponden a todo el paradigma y se utilizan para formar los tiempos compuestos. Las segundas son las únicas que reciben interpretación y que pueden aparecer en las construcciones existenciales (*hay, hubo, habrá petróleo en España*). 'Haber' con el sentido de posesión, muy común en español antiguo, sólo se conserva en unas pocas oraciones 'petrificadas' (*habemos Papa*). Por supuesto, este caso no se espera en NLQ.

## Análisis de los verbos auxiliares

A continuación veremos el comportamiento sintáctico de cada uno de estos verbos y su tratamiento dentro del sistema. Posteriormente, hablaremos de su interpretación

La gramática de análisis utiliza en sus reglas rasgos compuestos por un atributo y un valor. Este recurso formal está ampliamente extendido en el campo del procesamiento de lenguas naturales y de la lingüística, por lo que simplemente nos limitaremos a señalar sus características esenciales dentro de nuestro sistema:

- Todos los *constructos* (o categorías gramaticales) tienen que tener definidos los rasgos que pueden llevar.
- Hay tres tipos de rasgos, dependiendo de los valores que puedan tomar:
  1. *rasgos lógicos*: son de tipo binario, sólo aceptan como valor '0' ó '1'.
  2. *rasgos de 'caso'*: los valores son grupos de caracteres alfabéticos, que se utilizan para representar la estructura argumental de los verbos. Por ejemplo, los verbos transitivos llevan el rasgo TYP = NA, N por nominativo, A por acusativo.
  3. *rasgos numéricos*: los valores son números.

Los dos rasgos pertinentes para la tipología verbal son TYP (ya hemos visto que se trata de un tipo especial de rasgo) y AUX (es un rasgo numérico). Los verbos plenos llevan exclusivamente el rasgo TYP, para indicar qué tipo de complementos pueden llevar. Los verbos auxiliares, en cambio, llevan los dos rasgos pues uno sirve para señalar su función principal (TYP) y el otro para distinguir su función auxiliar (AUX). Se deduce de lo expuesto anteriormente que tanto 'ser' como 'estar' llevarán ambos rasgos; 'haber' tendrá sólo AUX en aquellas formas que funcionan como auxiliares, y además TYP en las que pueden funcionar también como existencial (terceras personas del singular).

Otros rasgos pertinentes para nuestra descripción son: LOC (rasgo lógico) para complementos locativos; TIM (lógico) para complementos temporales; y LAB (numérico) para los complementos preposicionales. En el último caso, a cada preposición le corresponde un número, que es el que se asigna como valor de LAB. De esta forma se sabe qué preposición rige el verbo en cuestión.

El algoritmo de parsing funciona de la siguiente manera (Zoepritz, 1984): las reglas que incorporan complementos toman el verbo principal (las reglas morfosintácticas se han encargado de formar los tiempos compuestos) y empiezan a buscar complementos a la derecha. Si el primer complemento (o bien un SN o bien un SP) es compatible con la estructura argumental del verbo (reflejada en el rasgo TYP) se crea un nuevo constructo (VC) con la combinación de rasgos pertinente. Para señalar el complemento añadido se le asigna el rasgo  $C_x = 1$ , donde x puede ser N(nominativo), A(acusativo).. y además se elimina la letra correspondiente de TYP. Por ejemplo, un verbo TYP = NDA que haya unido un dativo quedaría como TYP = NA y CD = 1. Si no encuentra todos los complementos requeridos busca a la izquierda del verbo hasta que completa la oración. Además, existen reglas para evitar que los complementos omitidos bloqueen el parsing, de tal forma que cualquier oración incompleta en cuanto a estructura argumental también es analizada.

Otra característica de la gramática es que los locativos y temporales son tratados de forma diferente a los SN y SP: estos complementos adverbiales (en nuestra terminología 'ablativos') no están incluidos en los valores de TYP, sino que cuentan con rasgos propios, como vimos más arriba. Los 'ablativos' se unen una vez que la estructura argumental está ya completa, esto es, cuando se ha creado el constructo SC. En principio, cualquier tipo de verbo permite ablativos<sup>3</sup>, por lo tanto si alguna construcción o verbo no permite tales complementos tiene que especificarse mediante los rasgos LOC y/o TIM con el valor '0'.

## 'Ser'

### Función principal

Este verbo tiene asignado TYP = NN porque exige dos SN nominativos, el sujeto y el predicado nominal. Hay que destacar que a efectos de concordancia sintáctica sólo es obligatoria la del sujeto con el verbo, aunque es mucho más habitual que ambos 'nominativos' lleven el mismo número:

Los minerales son un producto cotizado  
Los cetáceos son mamíferos acuáticos

Por otra parte, al igual que 'estar', 'ser' puede llevar como complementos adjetivos, SP y ablativos:

- adjetivos: ¿ *Es Gómez español ?*
- sintagmas preposicionales: ¿ *De quién es el cuadro más cotizado ?*
- locativos: ¿ *Dónde es la reunión de jefes de estado ?*

<sup>3</sup> Téngase en cuenta que a estas alturas del análisis el rasgo TYP ya no es pertinente pues se ha completado la

- temporales: *¿ Cuándo será la reunión de jefes de estado ?*

Dado que no existe ningún valor especial para los complementos adjetivos en el rasgo TYP; ni tampoco se ha considerado pertinente crear un TYP = NNP para un solo verbo (lo que implica reglas especiales), ambos complementos se tratan por medio de reglas *ad hoc* que únicamente se aplican cuando los verbos son 'ser' y 'estar'.

En cuanto a los locativos y temporales, ya hemos explicado que se aplican independientemente del TYP, por lo tanto estos complementos pasan por las mismas reglas que los verbos plenos.

### **Función auxiliar**

'Ser' se utiliza exclusivamente como auxiliar en las pasivas. Para distinguirse de los otros auxiliares lleva el rasgo AUX = 60 (el valor 60 es totalmente arbitrario, pero seguimos las convenciones dentro del formalismo USL). La regla que construye la forma compuesta pasiva copia todos los rasgos del verbo principal (el participio) y añade el rasgo PAS = 1 para indicar que el verbo solo puede pasar por las reglas específicas de la pasiva.

### **'Estar'**

#### **Función principal**

'Estar' presenta una diferencia radical con respecto a 'ser': no admite SN en la posición del predicado nominal. Esto se refleja en el valor de TYP mediante NZ. En la convención utilizada por las gramáticas USL, Z representa que sólo queda por unir al verbo el complemento indicado por la letra a la izquierda de Z. TYP = NZ significa que ese verbo únicamente admite un nominativo. Sería más correcto utilizar TYP = N para este verbo, pero a efectos prácticos de la gramática poner NZ permite utilizar algunas reglas escritas para 'ser'. En concreto, se usan las reglas en las que 'ser' incorpora el sujeto cuando ya se ha unido el predicado nominal. Esto se controla mediante el rasgo NOCPN = 1 que está asignado en la entrada de diccionario.

En cambio, al igual que 'ser', permite complementos adjetivos, preposicionales y ablativos, que se tratan de la misma manera y por las mismas reglas que 'ser':

- adjetivos: *¿ Está enfermo Sánchez ?*
- sintagmas preposicionales: *¿ Quién está de rector de la UAM ?*
- locativos: *¿ En qué continente está Dhomey ?*
- temporales: *¿ Cuándo estará el pedido ?*

#### **Función auxiliar**

¿ Se puede considerar que 'estar' es un auxiliar de pasiva ? No vamos a entrar en la polémica teórica (para conocer los detalles véase por ejemplo Marcos Marín (1980), cap. 13). Simplemente asumiremos que estas dos oraciones son equivalentes:

- ¿ Qué jugadores fueron seleccionados por Suárez ?
- ¿ Qué jugadores están seleccionados por Suárez ?

pues el pequeño matiz semántico que las separa no es relevante para consultar una base de datos. Ambas se interpretan como *¿ Qué jugadores seleccionó Suárez ?*

Como señalamos al comienzo, 'estar' es el verbo auxiliar para las perífrasis de gerundio. Para distinguirse de los otros auxiliares toma AUX = 65. La regla que forma la perífrasis exige que el primer verbo sea AUX = 65 y el segundo PPP = 1 (*present participle en USL*). El matiz de acción en

constructo nuevo los rasgos del verbo principal (menos PPR = 1) y toma la información morfosintáctica (persona, número, tiempo) del verbo auxiliar.

## 'Haber'

### Función auxiliar

Una de las funciones que realiza 'haber' es la de auxiliar de los tiempos compuestos. Se distingue con el rasgo AUX = 70 y por no llevar TYP. La regla que forma los tiempos compuestos toma los rasgos léxicos del participio y los rasgos morfosintácticos del auxiliar, teniendo en cuenta que si el auxiliar tenía PST (pasado) o PRS (presente) el nuevo constructo tendrá PST = 1 (*había exportado*, *ha exportado*); si el auxiliar estaba en futuro, el verbo compuesto llevará FUT = 1 (*habrá exportado*).

### Función principal

'Hay', 'había', 'hubo', 'habrá' son las formas especiales de 'haber' para las construcciones existenciales. Se diferencian del resto del paradigma en que se les asigna el rasgo TYP. Nótese, en cambio, que todas las formas de 'ser' y 'estar' pueden funcionar como auxiliar y principal. En 'haber' ambas funciones están especializadas.

Tradicionalmente las construcciones existenciales con 'hay' se han asimilado a oraciones impersonales como las de verbos de fenómenos naturales (véase el *Esbozo*, cap. 3). El nombre que acompaña al verbo se considera como un complemento de éste, y el sujeto queda indeterminado. Algunos autores (Seco, 1988) consideran que este complemento va en caso acusativo porque no concuerda con el verbo. En nuestro sistema, en cambio, le asignamos el rasgo TYP = NZ para indicar que requiere un único complemento nominal y además el rasgo NOCN = 1 que señala que debe pasar por las reglas que incorporan exclusivamente el predicado nominal.

Vamos pues a resumir la estrategia de análisis de la gramática: los tres verbos pasan por el mismo conjunto de reglas que se diferencian de las reglas para verbos plenos por el valor de TYP. El verbo auxiliar que acepta mayor número de complementos es 'ser' (TYP = NN), mientras que 'estar' y 'haber' admiten sólo uno de los dos argumentos (TYP = NZ). Por lo tanto, existe una regla que une el sujeto a un verbo con TYP = NN o TYP = NZ y NOCPN = 1, en la que hay comprobación de concordancia. Tras la aplicación de esta regla, el SV queda marcado con el rasgo CN = 1. La regla que incorpora predicados nominales se aplica sobre verbos con TYP = NN o TYP = NZ y NOCN = 1. En esta regla no hay comprobación de la concordancia de número. Dado que el verbo 'estar' está marcado con NOCPN = 1 en el diccionario, las reglas que incorporan predicados nominales nunca podrá aplicársele. De forma análoga, la presencia del rasgo NOCN = 1 en las formas 'hay', 'habrá', etc evita la aplicación de las reglas de incorporación de nominativos. Nótese además que el resultado de la aplicación de cualquiera de las dos reglas sobre una forma del verbo 'ser' coincide con la descripción estructural que se necesita para la aplicación de la otra regla. Por otra parte, el rasgo C = 0 o CPN = 0 se exige en las reglas que incorporan los correspondientes complementos para evitar la recursividad. La siguiente tabla resume todo el proceso.

	SER	ESTAR	HAY
Rasgos asignados en el diccionario	TYP = NN CN = 0 CPN = 0	TYP = NZ CN = 0 NOCPN = 1	TYP = NZ NOCN = 1 CPN = 0
Descripción estructural	Aplicación de la regla		Resultado
( Ser ) TYP = NN	Incorporación del sujeto		TYP = NZ CN = 1 CPN = 0
( Estar o Ser ) TYP = NZ NOCPN = 1 CN = 0			CN = 1 NOCPN = 1
( Ser ) TYP = NN	Incorporación del pred. nominal		TYP = NZ CPN = 1 CN = 0
( Hay o Ser ) TYP = NZ NOCN = 1 CPN = 0			CPN = 1 NOCN = 1

Figura 2. Incorporación de complementos

### Interpretación de los verbos auxiliares

Como dijimos anteriormente, no consideramos relevantes dentro del ámbito NLQ las distinciones semánticas entre 'ser' y 'estar' del tipo "intemporalidad / temporalidad" (Marcos Marin, 1980) o "existencia / estado" (R. Seco, 1988). No queremos decir con esto que no sean relevantes en otros contextos (traducción automática, por ejemplo) o en el uso cotidiano del castellano. Tradicionalmente se dice que ambos verbos son *cópulas* cuando enlazan un sujeto y un predicado nominal, y se les considera meros lazos de unión entre los auténticos elementos interpretativos (a menudo se pone el ejemplo del ruso y del árabe, donde el verbo copulativo no se expresa en determinados casos).

Por otra parte, en lógica de predicados oraciones como *algunos empleados están casados*, *algunos empleados son casados*, *hay empleados casados*, y *existen empleados casados* se analizan de la misma forma, empleando un cuantificador existencial (Garrido, 1988):

$$\exists(x) [ A(x) \& B(x) ]$$

que se lee: "existe por lo menos un x para el que se cumple que x es *empleado* y que x es *casado*".

En nuestro sistema se utiliza este tipo de interpretación, que considera que las construcciones con 'ser', 'estar' y 'hay' son equivalentes cuando se busca el valor de verdad acerca de la existencia de los predicados A y B. En realidad, la interpretación procede en dos pasos: en primer lugar, se crea un predicado provisional de 'ser', 'estar' o 'hay' durante el proceso de análisis. Este predicado se elimina

*semánticos*. El segundo paso consiste en agrupar en una fórmula lógico-conceptual el sujeto con el predicado nominal o adjetivo, y los adverbios cuando sea necesario. Estas formas lógico-conceptuales están basadas en el cálculo de predicados, pero expresado mediante varios predicados PROLOG distintos según el tipo de pregunta, de tal forma que las oraciones del ejemplo darían lugar a una representación como ésta:

```
(1)
      existe(x1,
            ejemplo_de(Empleado,x1) &
            ejemplo_de(Casado,x1)).
```

La evaluación en PROLOG de un predicado de este tipo procede sustituyendo *x1* por sus posibles valores y comprobando si alguno de esos valores cumple las dos condiciones. Si encuentra un valor para el que se cumplen, el resultado será *cierto*, mientras que retornará *falso* o fallo en cualquier otro caso. Por lo tanto, este predicado nos proporcionará información acerca del valor de verdad de nuestra pregunta, pero no acerca de los valores de la variable para los que se cumple. Pero muy frecuentemente, el usuario estará interesado en conocer no sólo si hay empleados casados, sino también los nombres de esos empleados. En este caso hará preguntas del tipo *¿Qué empleados están casados?*, *Dame empleados que estén casados*. Para responder adecuadamente a estas peticiones la interpretación se modifica, dando lugar a la siguiente forma lógico-conceptual:

```
(2)
      conjunto(x2,
              ejemplo_de(Empleado,x2) &
              ejemplo_de(Casado,x2)).
```

cuya evaluación en PROLOG proporcionará todos los valores de la variable *x2* para los que se cumplen ambas condiciones.

Un caso especial es aquel en el que el usuario no desea conocer el valor exacto de las variables sino simplemente cuántos casos aparecen en la base de datos con unas condiciones determinadas. Las posibles preguntas serían *¿Cuántos empleados están casados?*, *¿Cuántos empleados casados hay?* y su interpretación incluiría un nuevo predicado cuyo resultado es el número de veces en los que la evaluación tuvo éxito:

```
(3)
      cuenta(x3,
            ejemplo_de(Empleado,x3) &
            ejemplo_de(Casado,x3)).
```

Puede también darse el caso de que la pregunta incluya los determinantes *todos*, *cada*, p.ej. *¿Están casados todos los empleados?* La interpretación de esta oración debe incluir entonces el cuantificador universal en lugar del existencial:

```
(4)
      todo(x4,
           ejemplo_de(Empleado,x4) ->
           ejemplo_de(Casado,x4)).
```

Este predicado sólo dará *cierto* como resultado si para todos los empleados se cumple que están casados. 'Hay' no puede aparecer en construcciones de este tipo, dado su valor estrictamente existencial.

Aunque en los ejemplos anteriores los verbos actúan como nexos entre dos nombres o un nombre y un adjetivo, la interpretación no difiere sustancialmente cuando se trata de sintagmas preposicionales o ablativos. Por ejemplo, la forma lógico-conceptual que corresponde a *¿Qué inflación hubo en España en 1989?* sería:

```
(5) conjunto(x5,
      ejemplo_de("inflación",x5) &
      existe(x6,
            nombre("España",x6) &
            existe(x7,
                  ejemplo_de(fecha,1989) &
                  lugar(x5,x6) &
                  tiempo(x5,x7))))).
```

## Paráfrasis de la forma lógico-conceptual

Como adelantábamos en la introducción el sistema brinda al usuario la posibilidad de comprobar si su pregunta ha sido interpretada correctamente o, en el caso de que haya más de una interpretación, cuál de ellas es la solicitada. Con este fin, existe un generador que convierte la forma lógico-conceptual en una expresión en lengua natural. Esta paráfrasis no debe reproducir la pregunta de entrada, sino la interpretación que de esa pregunta hizo el sistema, por lo cual en ocasiones la apariencia de la paráfrasis es bastante diferente a la de la pregunta original.

La forma lógico-conceptual se traduce a una lista de categorías con rasgos. Sobre esta lista opera la gramática de generación. En el caso de oraciones con verbos auxiliares la lista no incluye ninguna categoría 'verbo', por lo que en muchos casos las paráfrasis son simplemente SNs.

En esta continuación damos ejemplos de paráfrasis de las formas lógicas que hemos visto más arriba:

### (1) existe

Paráfrasis: ¿ Hay empleados que son casados ?

### (2) conjunto

Paráfrasis: Busca empleados que son casados.

### (3) cuenta

Paráfrasis: Busca el número de empleados que son casados.

### (4) todo

Paráfrasis: Averigua si todos los empleados son casados.

### (5) ablativos

Paráfrasis: Busca inflaciones en un país llamado España en una fecha que es 1989.

---

## Bibliografía

1. **Garrido, J.** (1988): *Lógica y Lingüística*, Madrid, Síntesis.
2. **Gazdar, G. y Mellish, C.** (1989): *Natural Language Processing in PROLOG*, Wokingham, etc Addison - Wesley.
3. **Marcos Marín, F.** (1980): *Curso de Gramática Española*, Madrid, Cincel-Kapelusz.
4. **McCawley** (1981): *Everything that linguists have always wanted to know about logic*, Blackwell Oxford.
5. **Real Academia Española** (1973): *Esbozo de una nueva gramática de la lengua española*, 12ª reimp. Madrid, Espasa-Calpe.
6. **Seco, R.** (1988): *Manual de Gramática Española*, 11ª ed., Madrid, Aguilar.
7. **Sopeña, Luis** (1982): *Grammar of Spanish for User Specialty Languages*, Heidelberg, IBM Germany.
8. **Zoeppritz, M.** (1984): *Syntax for German in the User Specialty Languages System*, Tübingen, Ma: Niemeyer.